

## Program szkolenia:

# Projektowanie, tworzenie i testowanie aplikacji mobilnych na platformie Android

## Informacje:

<b>Nazwa:</b>	<b>Projektowanie, tworzenie i testowanie aplikacji mobilnych na platformie Android</b>
<b>Kod:</b>	<b>android-core</b>
<b>Kategoria:</b>	Android
<b>Grupa docelowa:</b>	developerzy
<b>Czas trwania:</b>	4 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

Szkolenie zostało przygotowane z myślą o programistach Java pragnących rozpocząć pracę na platformie Java oraz rozwinąć kompetencje do poziomu biegłego.

Szkolenie zawiera szereg praktycznych rozwiązań opracowanych na podstawie doświadczeń w dojrzałych projektach.

Program szkolenia został rozszerzony o aspekty architektury, wzorców i testowania automatycznego.

## Materiały wstępne

Przed szkoleniem możesz zapoznać się z serią naszych artykułów: [Zaawansowane programowanie na platformie Android](#).

## Zalety szkolenia:

- Aspekty wydajności
- Architektura i wzorce projektowe
- Typowe pułapki oraz najlepsze praktyki

## Szczegółowy program:

### 1. Podstawy systemu Android.

- 1.1. Historia systemu.
- 1.2. Podstawy budowy platformy Android.
- 1.3. Architektura systemu - podstawowe komponenty.
- 1.4. Komunikacja pomiędzy komponentami aplikacji.
- 1.5. Struktura projektu aplikacji.
- 1.6. Komunikacja pomiędzy aplikacjami zainstalowanymi w systemie.

### 2. Architektura aplikacji.

- 2.1. Wzorzec Mediator.
- 2.2. Layers.
- 2.3. Ports and Adapters.

### 3. Zasoby aplikacji androidowych.

- 3.1. Przegląd rodzajów zasobów.
- 3.2. Sposoby przechowywania grafiki.
- 3.3. Odwoływanie się do zasobów.
- 3.4. Uzależnienie zasobów od konfiguracji urządzenia.

### 4. Optymalizacja aplikacji i poszukiwanie błędów z użyciem narzędzi deweloperskich.

- 4.1. Przegląd narzędzi deweloperskich dostarczonych z Android SDK.

### 5. Projektowanie i implementacja warstwy prezentacji aplikacji.

- 5.1. Budowanie interfejsu użytkownika za pomocą widoków.
  - 5.1.1. Deklaratywne budowanie interfejsu użytkownika.
  - 5.1.2. Budowanie widoków z użyciem kodu w Javie.
  - 5.1.3. Obsługa cyklu życia widoków.

5.1.4. Projektowanie czytelnego interfejsu użytkownika.

5.1.5. Obsługa zdarzeń interfejsu użytkownika.

5.2. Budowa ekranów interfejsu użytkownika z wykorzystaniem Activity.

5.2.1. Implementacja Activity.

5.2.2. Obsługa cyklu życia.

5.2.3. Uruchamianie czynności.

5.2.4. Projektowanie przepływu (flow) w aplikacji.

5.2.5. Obsługa stosu Activity.

5.2.6. Zapisywanie i odtwarzanie stanu aplikacji.

5.3. Budowa interfejsów użytkownika na wiele typów urządzeń (smartphone, tablet, tv) z wykorzystaniem Fragmentów.

5.3.1. Tworzenie i używanie Fragmentów.

5.3.2. Obsługa cyklu życia Fragmentów.

5.3.3. Manipulacja Fragmentami.

5.3.4. Komunikacja pomiędzy Fragmentami a Activity.

5.4. Używanie widoków opartych na adapterach w celu zaprezentowania na ekranie dużej ilości danych.

5.4.1. ListView

5.4.2. GridView

5.4.3. Spinner

5.5. Interakcja użytkownika z aplikacją za pomocą Menu i Akcji "podręcznych".

5.5.1. Tworzenie i obsługa Menu.

5.5.2. Używanie ActionBar.

5.5.3. Sposoby nawigacji po aplikacji z użyciem ActionBar.

5.6. Informowanie użytkownika o zdarzeniach za pomocą okien dialogowych i powiadomień (Notifications)

5.6.1. Tworzenie okien dialogowych.

5.6.2. Komunikacja dialogów z Activity.

5.6.3. Użycie powiadomień systemowych.

5.7. Stylowanie wyglądu aplikacji.

5.7.1. Deklaratywna definicja stylów.

5.7.2. Rozszerzanie stylów znajdujących się na urządzeniu.

5.8. Rysowanie interfejsu użytkownika oraz obsługa elementów graficznych i animacji.

5.8.1. Sposoby rysowania interfejsu użytkownika.

5.8.2. Tworzenie deklaratywnych zasobów graficznych.

5.8.3. Tworzenie skalowalnych zasobów graficznych (9 patch).

5.8.4. Wczytywanie obrazów z zewnętrznych źródeł.

5.8.5. Tworzenie zasobów animacji.

5.8.6. Uruchamianie i obsługa animacji.

5.9. Tworzenie interfejsu do zarządzania ustawieniami użytkownika.

5.10. Obsługa asynchronicznych operacji w warstwie prezentacji.

## **6. Trwały zapis danych użytkownika.**

6.1. Użycie SharedPreferences do zapisu ustawień użytkownika.

6.2. Obsługa plików w systemie Android.

6.3. Zapis i odczyt danych z External i Internal Storage.

6.4. Użycie bazy danych SQLite do zarządzania danymi użytkownika.

6.4.1. Tworzenie relacyjnych baz danych.

6.4.2. Tworzenie i wykonywanie zapytań do bazy danych i obsługa ich parametrów.

6.5. Zarządzanie danymi aplikacji z użyciem Content Providera.

6.5.1. Sposoby dostępu do danych.

6.5.2. Obsługa nośnika danych - Cursora.

6.5.3. Dostęp do danych publikowanych przez inne aplikacje.

6.5.4. Tworzenie warstwy dostępu do danych.

6.5.5. Implementacja bazy SQLite jako źródła danych.

6.5.6. Obserwowanie zmian danych i aktualizacja interfejsu użytkownika w przypadku wystąpienia zmian.

6.5.7. Testowanie warstwy dostępu do danych.

6.6. Asynchroniczny dostęp do danych za pomocą Loaderów.

## **7. Wykonywanie długotrwałych operacji w tle.**

7.1. Tworzenie usług (Service) wykonujących czynności w tle.

7.2. Uruchamianie i zarządzanie usługami.

7.3. Obsługa cyklu życia usługi.

7.4. Tworzenie usług wielowątkowych.

7.5. Powiadomianie użytkownika o rezultacie wykonywanych w tle operacji.

## **8. Analiza obsługi procesów i wątków w systemie Android.**

8.1. Tworzenie aplikacji, które są płynne w użyciu dzięki odpowiedniemu zarządzaniu wątkami.

## **9. Bezpieczeństwo systemu i aplikacji.**

9.1. Analiza modelu bezpieczeństwa Androida.

9.2. Używanie pozwoleń (Permissions) w celu uzyskania dostępu do zasobów urządzenia.

9.3. Ochrona dostępu do komponentów aplikacji.

9.4. Definicja i używanie własnych pozwoleń.

## **10. Komunikacja aplikacji z usługami zewnętrznymi.**

10.1. Architektura systemów rozproszonych.

10.2. Łączenie się z usługami zewnętrznymi za pomocą socketów.

10.3. Komunikacja z usługami zewnętrznymi z użyciem protokołu HTTP i serializacji JSON.

## **11. Utrzymanie kompatybilności ze starszymi urządzeniami.**

11.1. Użycie biblioteki kompatybilności wstecznej support-v4.

11.2. Użycie zewnętrznej biblioteki implementującej ActionBar na starszych urządzeniach - ActionBarSherlock.

## **12. Publikacja aplikacji w Google Play.**

12.1. Proces optymalizacji aplikacji.

12.2. Podpisywanie aplikacji.

12.3. Umieszczanie aplikacji w sklepie Google Play.

## **13. Testowanie automatyczne.**

13.1. Testy jednostkowe.

13.2. Testy akceptacyjne.